

Upstream Liability Research
Upstream Liability Workgroup
Commission on Misclassification & Payroll Fraud
August 11, 2021

New York Bill: [S2766C](#) shifts liability to contractors for unpaid wages owed to a worker by any subcontractor on any level of a project. (Subcontractors not liable for unpaid workers upstream or downstream of themselves)

- Article [here](#)
- Status: Passed Assembly and Senate, now awaiting Gov. Cuomo's signature as of 8/2.

California Bill/Laws:

- **[Labor Code section 2810.3](#)**: Makes an entity that uses a subcontractor liable for the violations of that sub. Requires that a client employer who obtains workers through a subcontractor share in liability for any wage and workers compensation issues. Client employers cannot shift all liability for wage and workers' compensation violations to the subcontractor.
- **[Labor Code Section 218.7](#)**: Makes general contractors (or direct contractors) liable for subcontractors' failure to pay wage, fringe, or other benefit payments or contributions (at any tier) including subcontractors' failure to pay any interest incurred on such wages, fringe, or other benefits, but does not extend to penalties or liquidated damages.
 - See rest of summary [here](#)
- **[SB 1399](#)**: included a provision that would have made fashion brands and retailers that contract to have clothing manufactured for them liable for any wage theft, even if workers were shortchanged by third-party subcontractors.
 - "These so-called retailers have frustrated the law, avoiding liability for this systemic abuse, by creating layers of subcontracting, which has enabled them to claim that they do not fall under the definition of "garment manufacturer," as defined in AB 633, and are therefore not liable for these egregious wage violations. The intent of AB 633 must be restored, and **upstream liability** established, or the unrelenting problem of wage theft in the garment industry will continue."

Maryland Law: under 2021 [SB 947](#), a general contractor on a project for construction services under a contract entered into after 9/30/2018, is jointly and severally liable for a violation of this subtitle that is committed by a subcontractor, regardless of whether the subcontractor is in a direct contractual relationship with the general contractor.

- Status: in Senate

Minnesota Law: [section 176.215, subdivision 1](#) (2016) states: "Where a subcontractor fails to comply with this chapter, the general contractor, or intermediate contractor, or subcontractor is

liable for payment of all compensation due an employee of a subsequent subcontractor who is engaged in work upon the subject matter of the contract.”

- If the employee of a subcontractor is injured and the subcontractor does not have workers’ compensation insurance, the general contractor is liable for the injured worker’s benefits. If there is a chain of multiple contractors, the employee’s workers’ compensation benefits may be collected from any contractor up the chain.
- See infosheet on contractor liability [here](#)

New Jersey Law:

- 2019 [Wage Theft Act \(WTA\)](#) includes a provision which makes client-employers and labor contractors jointly and severally liable “for any violations of the provisions of State wage and hour laws,” including those on retaliation. In fact, the WTA declares any waiver of its “joint and several liability” section “void and unenforceable.”
 - [Bill text](#): A client employer and a labor contractor providing workers to the client employer shall be subject to joint and several liability and shall share civil legal responsibility for any violations of the provisions of State wage and hour laws or violations of the provisions of section 10 of P.L.1999, c.90

[Public Task Forces Take on Employee Misclassification: Best Practices](#)

Policy & Data Brief from National Employment Law Project

- “Since 2007, more than half the states have formed interagency task forces to take on misclassification. In 2018 and 2019, there was a task force resurgence, with eight new or revitalized task forces.”
- See full report [here](#)